UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/810,716	03/16/2001	Hiang-Swee Chiang	3996-4002US1	2184

7590 07/07/2004

MORGAN & FINNEGAN, L.L.P.
345 Park Avenue
New York, NY 10154-0053

EXAMINER	
WOOD, WILLIAM H	

ART UNIT	PAPER NUMBER
2124	

DATE MAILED: 07/07/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/810,716	CHIANG, HIANG-SWEE	
	Examiner	Art Unit	
	William H. Wood	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 March 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-166* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-166* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *16 March 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Claims 1-166 are pending and have been examined.

Priority

1. Applicant's claim for domestic priority under 35 U.S.C. 119(e) is acknowledged.

Information Disclosure Statement

2. The information disclosure statement (IDS) submitted on 16 March 2001, 12 July 2001 and 31 January 2002 were considered by the examiner.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 1, 79, 157 and 162 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 1, 79, 157 and 162 are rejected under 35 U.S.C. 112, second paragraph, as being incomplete for omitting essential steps, such omission amounting to a gap between the steps. See MPEP § 2172.01. The omitted steps are (as found in Applicant's specification on page 4, line 23): *if the application framework code is not available for the web application* (not included portion), *then the web application server generates the application framework code* (included portion).

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1-166 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Lau** (USPN 5,987,247) in view of **Lindhorst** et al. (USPN 6,337,696) in further view of Admitted Prior Art (**APA**, Applicant's background specification).

Claim 1

Lau disclosed a method of generating computer code for application *(column 5, lines 33-40)*, comprising:

- receiving input files, wherein the input files are at least one application *(column 3, lines 14-16; column 5, line 35)*;
- determining if an application framework code is available for the web application *(column 6, lines 21-22; and column 5, lines 46-50; must determine if saved framework exists for future editing/changing)*;
- generating the application framework code *(column 5, lines 33-40)*, a business logic foundation code *(column 6, lines 24-29)*, an event handler skeleton *(column 5, lines 33-40; column 13, lines 28-44)* and a user interface code *(column 5, lines 33-40)*;
- receiving application business logic objects from a web developer *(column 6, lines 24-29)*;
- receiving methods *(column 13, lines 28-44)*;

- organizing the application framework code, the application business logic objects and the methods into application source code *(column 4, lines 25-27, linking and compiling, along with building and preparing the code)*;
- compiling the application source code *(column 4, lines 25-27)*;

Lau did not explicitly state generating code for a *web* application. **Lindhorst** demonstrated that it was known at the time of invention to generate code for web applications (column 2, lines 11-19; column 3, lines 1-4; and column 4, lines 10-16) using, among other elements, a graphical interface input file (column 11, lines 37-40). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the application framework generation system of **Lau** with graphical design input for the web as found in **Lindhorst**'s teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide less technical and thus easier methods, such as frameworks and automatic code generation, for average users to program in various known environments, like the web (**Lindhorst**: column 3, lines 37-45; **Lau**: column 2, lines 43-47).

Lau did not explicitly state *receiving event handler methods*. **Lindhorst** demonstrated that it was known at the time of invention to provide event handler methods (column 3, lines 16-20; column 11, line 66 to column 12, line 17; figure 6). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the code generation system of **Lau** with provided event handler methods as found in **Lindhorst**'s teaching. This implementation would have been obvious because one of ordinary skill

in the art would be motivated to free a user from being required to know complex technical details of programming, thus making it easier (**Lindhorst**: column 3, lines 38-45).

Lau and **Lindhorst** did not explicitly state *compiling/binding the web application source code with the input files at runtime*. **Lau** demonstrated that it was known at the time of invention to implement using JAVA (column 6, lines 34). **APA** demonstrated it was known at the time of invention to bind at runtime (page 3, lines 2-3; interpretation and thus binding). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the framework development system of **Lau** and **Lindhorst** with run-time binding of various files/inputs as suggested by JAVA's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to allow for changes/improvements right up until actual use of code. Finally, runtime interpretation/binding indicates allowing updated/modified files up to runtime.

Claim 2

Lau disclosed a method of generating computer code for a application *(column 5, lines 33-40)*, comprising:

- receiving input files, wherein the input files are at least one application *(column 3, lines 14-16; column 5, line 35)*;

- generating an application framework code *(column 5, lines 33-40)* and event handler skeleton *(column 5, lines 33-40; column 13, lines 28-44)*;
- receiving application business logic objects *(column 6, lines 24-29)*;
- organizing the application framework code, the application business logic objects and the into application source code *(column 4, lines 25-27, linking and compiling, along with building and preparing the code)*; and

Lau did not explicitly state generating code for a *web* application. **Lindhorst** demonstrated that it was known at the time of invention to generate code for web applications (column 2, lines 11-19; column 3, lines 1-4; and column 4, lines 10-16) using, among other elements, a graphical interface input file (column 11, lines 37-40). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the application framework generation system of **Lau** with graphical design input for the web as found in **Lindhorst**'s teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide less technical and thus easier methods, such as frameworks and automatic code generation, for average users to program in various known environments, like the web (**Lindhorst**: column 3, lines 37-45; **Lau**: column 2, lines 43-47).

Lau did not explicitly state *receiving event handler methods*. **Lindhorst** demonstrated that it was known at the time of invention to provide event handler methods (column 3, lines 16-20; column 11, line 66 to column 12, line 17; figure 6). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the code

generation system of **Lau** with provided event handler methods as found in **Lindhorst**'s teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to free a user from being required to know complex technical details of programming, thus making it easier (**Lindhorst**: column 3, lines 38-45).

Lau and **Lindhorst** did not explicitly state *binding the web application source code with the input files at runtime*. **Lau** demonstrated that it was known at the time of invention to implement using JAVA (column 6, lines 34). **APA** demonstrated it was known at the time of invention to bind at runtime (page 3, lines 2-3; interpretation and thus binding). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the framework development system of **Lau** and **Lindhorst** with run-time binding of various files/inputs as suggested by JAVA's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to allow for changes/improvements right up until actual use of code.

Claim 3

Lau and **Lindhorst** disclosed the method of claim 2, wherein generating an event handler skeleton further comprises:

- parsing at least one input file *(**Lindhorst**: column 11, lines 37-40)*;
- reviewing the parsed input file for a tag type, an attribute name and an attribute value *(**Lindhorst**: column 13, lines 22-64)*; and

- determining an event handler method based on the tag type, the attribute name and the attribute value *(**Lindhorst**: column 13, lines 22-64)*.

Claim 4

Lau and **Lindhorst** disclosed the method of claim 2, wherein the web application source code is generated in an object-oriented programming language *(**Lau**: column 6, line 34)*.

Claim 5

Lau and **Lindhorst** disclosed the method of claim 4, wherein the object-oriented programming language is Java *(column 6, line 34)*.

Claim 6

Lau and **Lindhorst** disclosed the method of claim 4, wherein the object-oriented programming language is C++ *(column 3, line 65)*.

Claim 7

Lau and **Lindhorst** disclosed the method of claim 2, further comprising determining if the application framework code is available for the web application *(**Lau**: column 6, lines 21-22; and column 5, lines 46-50; must determine if saved framework exists for future editing/changing)*.

Claim 8

Lau and **Lindhorst** disclosed the method of claim 2, further comprising generating a business logic foundation code *(**Lau**: column 6, lines 24-29)*.

Claim 9

Lau and **Lindhorst** disclosed the method of claim 2, further comprising generating a graphical user interface code *(**Lau**: column 5, line 39)*.

Claim 10

Lau and **Lindhorst** disclosed the method of claim 9, wherein generating a graphical user interface code is based on the input files *(**Lau**: column 5, lines 33-39; design; column 4, lines 11-28)*.

Claim 11

Lau and **Lindhorst** disclosed the method of claim 2, wherein generating an event handler skeleton is based on the input files *(**Lau**: column 5, lines 33-39; design; column 4, lines 11-28)*.

Claim 12

Lau and **Lindhorst** disclosed the method of claim 2, further comprising compiling the web application source code *(**Lau**: column 4, lines 25-27)*.

Claim 13

Lau and **Lindhorst** did not explicitly state the method of claim 2, further comprising interpreting the web application source code. **Lau** demonstrated that it was known at the time of invention to implement using JAVA (column 6, lines 34). Official Notice is taken that Java technology is known to include a interpretation system. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the framework development system of **Lau** and **Lindhorst** with interpreting code as suggested by JAVA's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide a system of easy programmability (interpretation can be changed quickly on the fly, especially useful in system testing).

Claim 14, 16-17

Lau and **Lindhorst** did not explicitly state the method of claim 2, wherein the input files are in XML, cHTML or WML format. **APA** demonstrated that it was known at the time of invention to utilize XML and WML (page 3, lines 13-14). Official Notice is take that cHTML was known at the time of invention. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the input files of **Lau** and **Lindhorst** with the above formats as found in **APA**'s teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide as many formats as possible in order to be of use to the largest community of developers possible and thus increase usefulness of the system.

Claim 15

Lau and **Lindhorst** disclosed the method of claim 2, wherein the input files are in HTML format *(**Lindhorst**: column 11, lines 37-40)*.

Claim 18

Lau and **Lindhorst** disclosed the method of claim 2, further comprising receiving modified input files *(see motivation under claim 2; runtime interpretation/binding indicates allowing updated/modified files up to runtime)*.

Claim 19

Lau and **Lindhorst** did not explicitly state the method of claim 18, further comprising compiling the modified input files at runtime. **Lau** demonstrated that it was known at the time of invention to implement using JAVA (column 6, lines 34). Official Notice is taken that Java technology is known to include a just-in-time compiling system (in other words compiling at run time). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the framework development system of **Lau** and **Lindhorst** with run-time compiling as suggested by JAVA's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to allow for changes/improvements right up until actual use of code (see claim 18).

Claim 20

Lau and **Lindhorst** disclosed the method of claim 19, further comprising binding the web application source code with the modified input files at runtime (see claim 2 above).

Claim 21

Lau and **Lindhorst** did not explicitly state the method of claim 20, wherein the modified input files are compiled into DOM objects at runtime *(APA: page 3, lines 14-16)*. **APA** demonstrated that it was known at the time of invention to compile mark up language files into DOM (page 3, lines 14-16). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the code development system of **Lau** and **Lindhorst** with DOM compilation as found in **APA**'s teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide a easily handled structure for development (**APA**: page 3, lines 16-22).

Claim 22

Lau and **Lindhorst** did not explicitly state the method of claim 18, further comprising interpreting the modified input files at runtime. **Lau** demonstrated that it was known at the time of invention to implement using JAVA (column 6, lines 34). Official Notice is taken that Java technology is known to include a interpretation system. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the framework development system of **Lau** and **Lindhorst** with interpreting code as suggested by JAVA's teaching. This implementation would have been obvious because

one of ordinary skill in the art would be motivated to provide a system of easy programmability (interpretation can be changed quickly on the fly, especially useful in system testing).

Claim 23

Lau and **Lindhorst** disclosed the method of claim 22, further comprising binding the web application source code with the interpreted modified input files at runtime *(see claim 2 and 22; further binding required in order for code to work correctly)*.

Claim 24

Lau and **Lindhorst** disclosed the method of claim 2, further comprising generating application runtime properties *(**Lau**: column 5, lines 39-40; attributes at least)*.

Claim 25

Lau and **Lindhorst** did not explicitly state the method of claim 2, further comprising generating application SQL statements. **Lau** demonstrated that it was known at the time of invention to utilize database management systems in business logic (column 8, lines 16-25). Official Notice is taken that SQL was known at the time of invention. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the code framework system of **Lau** with generating SQL as well. This implementation would have been obvious because one of ordinary skill in the art would

be motivated to provide **Lau**'s system with the ability to communicate with as many differing systems/environments as possible and thus increasing flexibility and usability.

Claim 26

Lau and **Lindhorst** disclosed the method of claim 2, wherein the application framework code comprises an application object and a servlet web application framework object *(column 5, lines 15-19).*

Claims 27-166

The limitations of system claims 27-166 correspond to the limitations of method claims 2-26 and as such are rejected in the same manner.

Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to William H. Wood whose telephone number is (703)305-3305. The examiner can normally be reached 7:30am - 5:00pm Monday thru Thursday and 7:30am - 4:00pm every other Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662. The fax phone numbers for the organization where this application or proceeding is assigned are (703)746-7239 for regular communications and (703)746-7238 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703)305-3900.

William H. Wood
June 27, 2004

Kakali Chaki

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100